

# Linguagens Formais e Autômatos

**P. Blauth Menezes**

blauth@inf.ufrgs.br

**Departamento de Informática Teórica  
Instituto de Informática / UFRGS**



# Linguagens Formais e Autômatos

P. Blauth Menezes

- 1 **Introdução e Conceitos Básicos**
- 2 **Linguagens e Gramáticas**
- 3 **Linguagens Regulares**
- 4 **Propriedades das Linguagens Regulares**
- 5 **Autômato Finito com Saída**
- 6 **Linguagens Livres do Contexto**
- 7 **Propriedades e Reconhecimento das Linguagens Livres do Contexto**
- 8 **Linguagens Recursivamente Enumeráveis e Sensíveis ao Contexto**
- 9 **Hierarquia de Classes e Linguagens e Conclusões**

# 5 – Autômato Finito com Saída

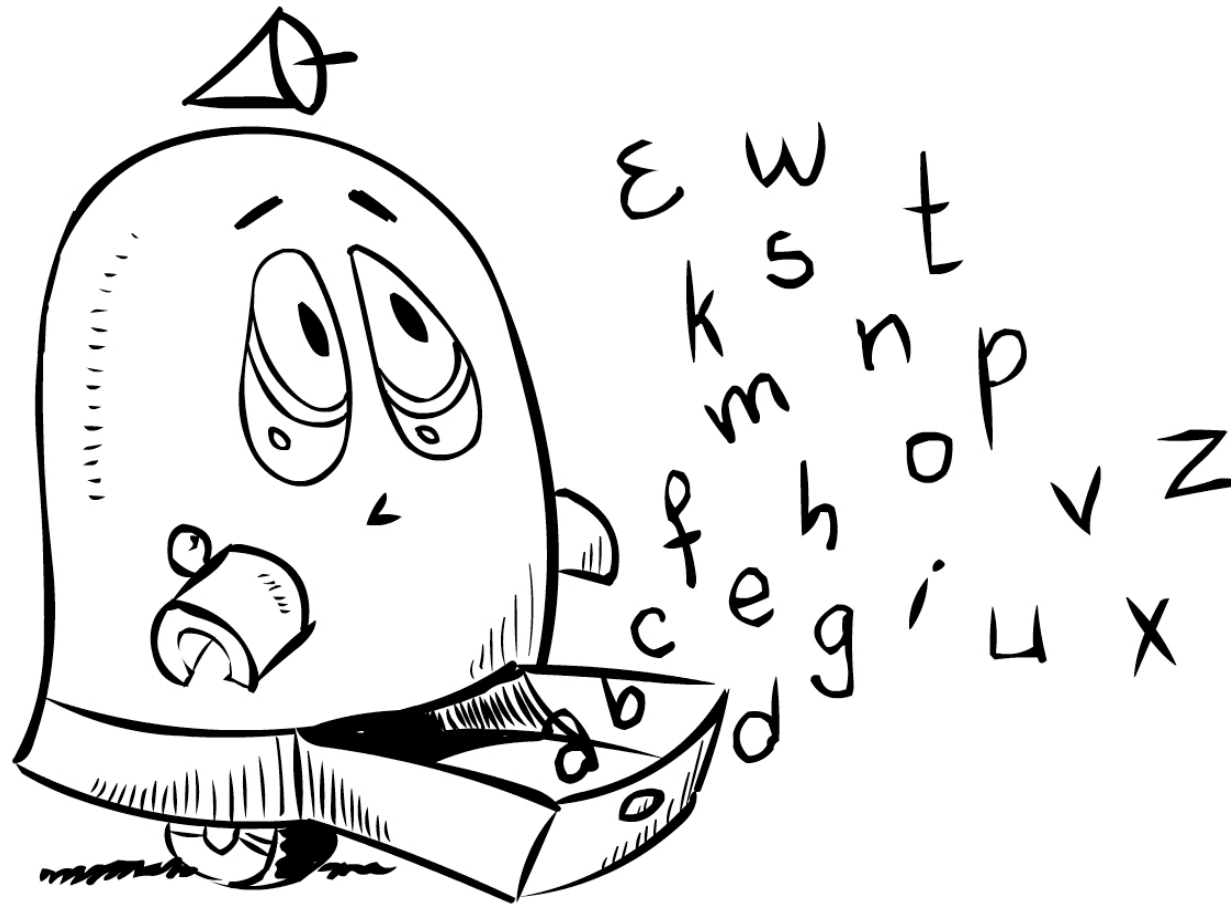
**5.1 Máquina de Mealy**

**5.2 Máquina de Moore**

**5.3 Equivalência das Máquina de Moore e Mealy**

**5.4 Hipertexto e Hiperímia como Autômato Finito com Saída**

**5.5 Animação como Autômato Finito com Saída**



# 5 – Autômato Finito com Saída

# 5 Autômato Finito com Saída

## ◆ Conceito básico de autômato finito

- aplicações práticas restritas
- informação de saída limitada à lógica binária aceita/rejeita

## ◆ Geração de uma palavra de saída

- estende a definição de Autômato Finito
- mesma classe de linguagens reconhecidas

## ◆ As saídas podem ser associadas

- às transições: Máquina de Mealy
- aos estados Máquina de Moore

## ◆ A saída não pode ser lida: não é memória auxiliar

- definida sobre um alfabeto especial: alfabeto de símbolos de saída
  - \* pode ser igual ao alfabeto de entrada
- saída: fita de saída, independente da de entrada
- cabeça da fita de saída
  - \* move uma célula para a direita a cada símbolo gravado
- resultado do processamento
  - \* estado final (condição de aceita/rejeita)
  - \* informação contida na fita de saída

## ◆ Máquinas de Mealy e Moore

- **modificações** sobre o AFD
- **exercício**
  - \* não-determinismo
  - \* movimentos vazios

## ◆ Aplicações dos autômatos finitos com saída

- tradicionais
  - \* analisador léxico
  - \* processador de textos ...
- **WWW (World Wide Web)**
  - \* **hipertexto e hipermídia**
  - \* **animação quadro-a-quadro**

# 5 – Autômato Finito com Saída

**5.1 Máquina de Mealy**

**5.2 Máquina de Moore**

**5.3 Equivalência das Máquina de Moore e Mealy**

**5.4 Hipertexto e Hipermissão como Autômato Finito com Saída**

**5.5 Animação como Autômato Finito com Saída**



# 5.1 Máquina de Mealy

## ◆ Para cada transição da máquina

- gera uma palavra de saída (pode ser vazia)

## Def: Máquina de Mealy

$$M = (\Sigma, Q, \delta, q_0, F, \Delta)$$

- $\Sigma$  - alfabeto (de símbolos) de entrada
- $Q$  - conjunto de estados (finito)
- $\delta$  - função programa ou função de transição (função parcial)

$$\delta: Q \times \Sigma \rightarrow Q \times \Delta^*$$

- $q_0$  - elemento distinguido de  $Q$ : estado inicial
- $F$  - subconjunto de  $Q$ : conjunto de estados finais
- $\Delta$  - alfabeto (de símbolos) de saída

### ◆ Máquina de Mealy $\times$ AFD

- $\Sigma$ ,  $Q$ ,  $q_0$  e  $F$  são como no AFD

## ◆ **Computação, para entrada $w$**

- sucessiva aplicação da função programa
- para cada símbolo de  $w$  (da esquerda para a direita)
- até ocorrer uma condição de parada

## ◆ **Palavra vazia como saída**

- nenhuma gravação é realizada
- não move a cabeça da fita de saída

## ◆ **Se todas as transições geram saída vazia**

- processa como se fosse um AFD

## ◆ **Definição formal da função programa estendida**

- exercício

## Exp: Máquina de Mealy: Diálogo

Aplicação comum e recomendada para os autômatos com saída

- projeto de diálogo entre um programa e o seu usuário
- determina, eventualmente, ações internas ao sistema

Diálogo pode ser de dois tipos

- comandado pelo programa
- comandado pelo usuário

## Exp: ...Máquina de Mealy: Diálogo

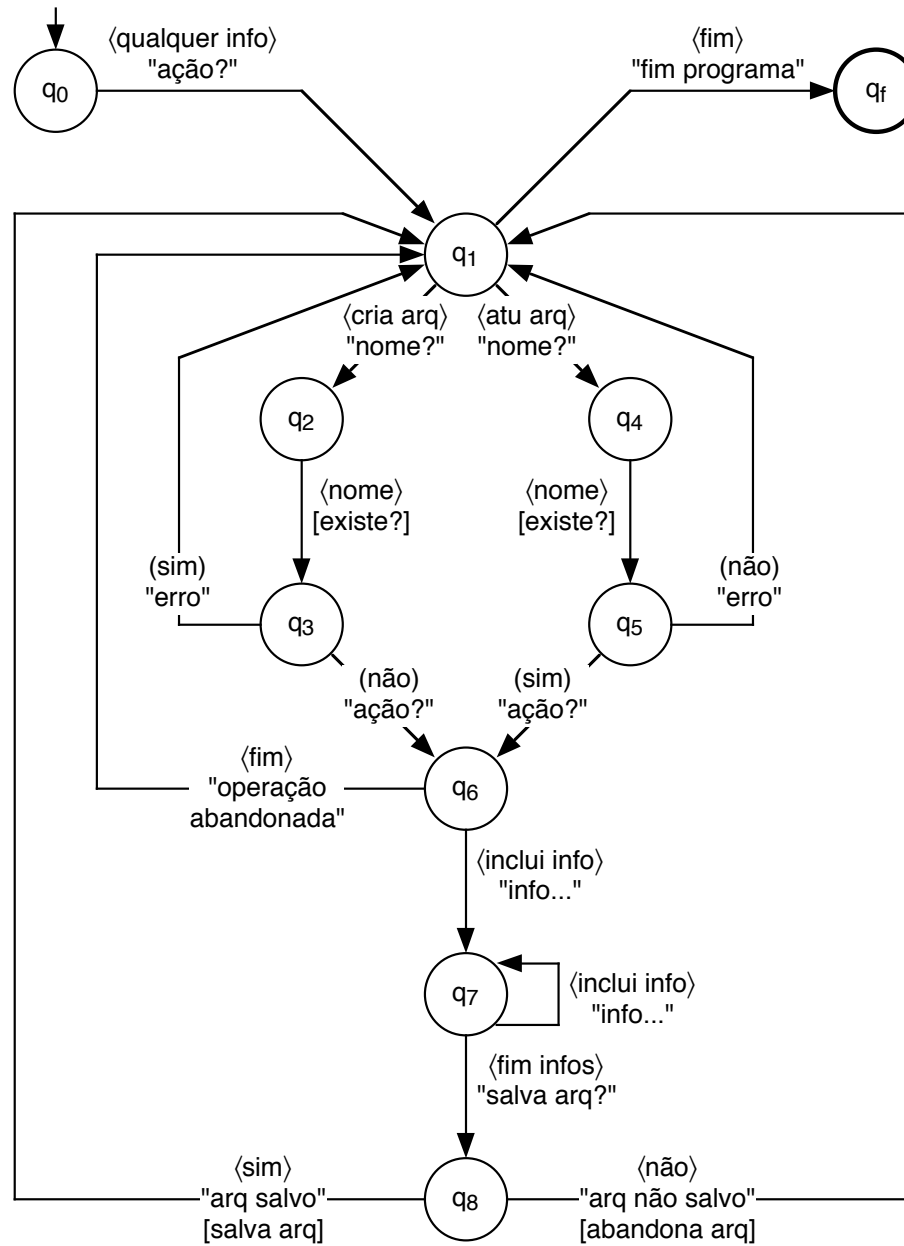
Exemplo de diálogo que cria e atualiza arquivos

- $\langle \dots \rangle$  entrada fornecida pelo usuário (em um teclado, por exemplo)
- $" \dots "$  saída gerada pelo programa (em um vídeo, por exemplo)
- $[ \dots ]$  ação interna ao programa (sem comunicação com o usuário)
- $( \dots )$  resultado de ação interna ao programa (entrada no diagrama)

Máquina de Mealy

$$M = (\Sigma, \{ q_0, q_1, \dots, q_8, q_f \}, \delta, q_0, \{ q_f \}, \Delta)$$

- $\Sigma$  e  $\Delta$ : símbolos (**palavras** do **português**) de entrada/saída válidos



# 5 – Autômato Finito com Saída

**5.1 Máquina de Mealy**

**5.2 Máquina de Moore**

**5.3 Equivalência das Máquina de Moore e Mealy**

**5.4 Hipertexto e Hipermissão como Autômato Finito com Saída**

**5.5 Animação como Autômato Finito com Saída**

## 5.2 Máquina de Moore

### ◆ Possui uma segunda função

- gera uma palavra de saída (pode ser vazia)
- para cada **estado** da máquina



## Def: Máquina de Moore

$$M = (\Sigma, Q, \delta, q_0, F, \Delta, \delta_S)$$

- $\Sigma$  - alfabeto (de símbolos) de entrada
- $Q$  - conjunto de estados (finito)
- $\delta$  - função programa ou função de transição (função parcial)

$$\delta: Q \times \Sigma \rightarrow Q$$

- $q_0$  - elemento distinguido de  $Q$ : estado inicial
- $F$  - subconjunto de  $Q$ : conjunto de estados finais
- $\Delta$  - alfabeto (de símbolos) de saída
- $\delta_S$  - função de saída (função total)

$$\delta_S: Q \rightarrow \Delta^*$$

## ◆ Máquina de Moore × AFD & Mealy

- $\Sigma$ ,  $Q$ ,  $\delta$ ,  $q_0$  e  $F$  são como no AFD
- $\Delta$  é como na Máquina de Mealy

## ◆ Computação, para entrada $w$

- sucessiva aplicação da função programa
  - \* para cada símbolo de  $w$  (da esquerda para a direita)
  - \* até ocorrer uma condição de parada
- *juntamente* com a sucessiva aplicação da função de saída
  - \* cada estado atingido

## ◆ Palavra vazia como saída

- nenhuma gravação é realizada
- não move a cabeça da fita de saída

## ◆ Se todas as transições geram saída vazia

- processa como se fosse um AFD

## ◆ Definição formal da função programa estendida

- exercício

# Exp: Máquina de Moore: Análise Léxica

## Analizador Léxico

- autômato finito (em geral, determinístico)
- **identifica** os **componentes básicos** da linguagem
  - \* números, identificadores, separadores, etc

## Máquina de **Moore** como Analizador Léxico

- cada **estado final**
  - \* **associado** a uma **unidade léxica**
  - \* a **saída** descreve ou **codifica** a unidade léxica identificada
- estados **não-finais**
  - \* em geral, **saída vazia**

# 5 – Autômato Finito com Saída

**5.1 Máquina de Mealy**

**5.2 Máquina de Moore**

**5.3 Equivalência das Máquina de Moore e Mealy**

**5.4 Hipertexto e Hipermissão como Autômato Finito com Saída**

**5.5 Animação como Autômato Finito com Saída**

## 5.3 Equivalência das Máquinas de Moore e de Mealy

### ◆ Equivalência

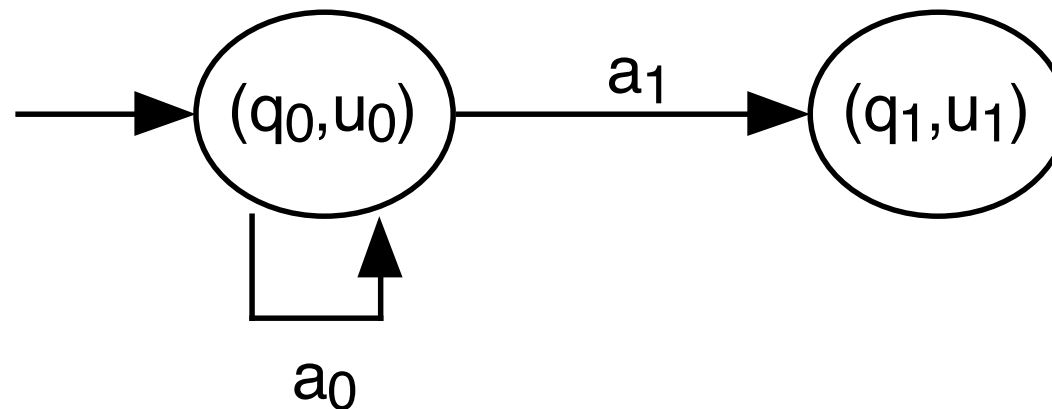
- não é válida para a entrada vazia (por quê?)
- demais casos
  - \* pode ser facilmente verificada

## Teorema: Máquina de Moore $\rightarrow$ Máquina de Mealy

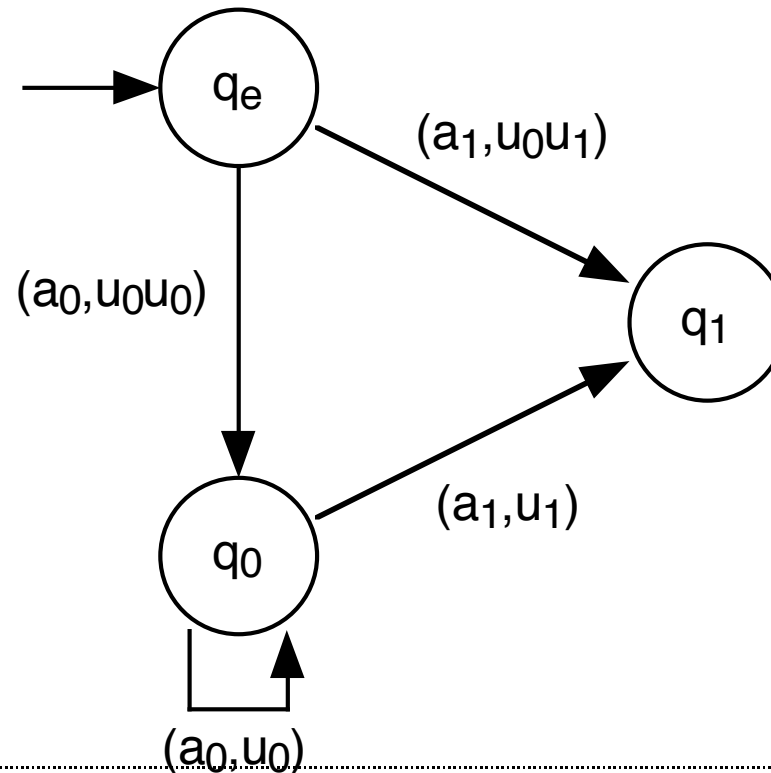
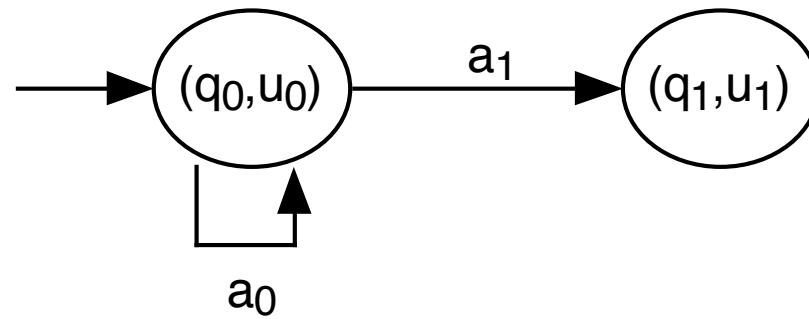
Toda Máquina de Moore pode ser simulada por uma Máquina de Mealy, para entradas não vazias

Prova: (por indução)

Supondo



Correspondente Mealy ???





$M = (\Sigma, Q, \delta, q_0, F, \Delta, \delta_S)$ , Máquina de Moore qualquer

Correspondente Mealy

$$ME = (\Sigma, Q \cup \{q_e\}, \delta_{ME}, q_e, F, \Delta)$$

Estado  $q_e$

- **referenciado** somente na **primeira transição** executada
- garante a geração da **saída** referente ao **estado inicial**  $q_0$  de Moore

Função programa  $\delta_{ME}$

- $\delta_{ME}(q_e, a) = (\delta(q_0, a), \delta_S(q_0) \delta_S(\delta(q_0, a)))$
- $\delta_{ME}(q, a) = (\delta(q, a), \delta_S(\delta(q, a)))$

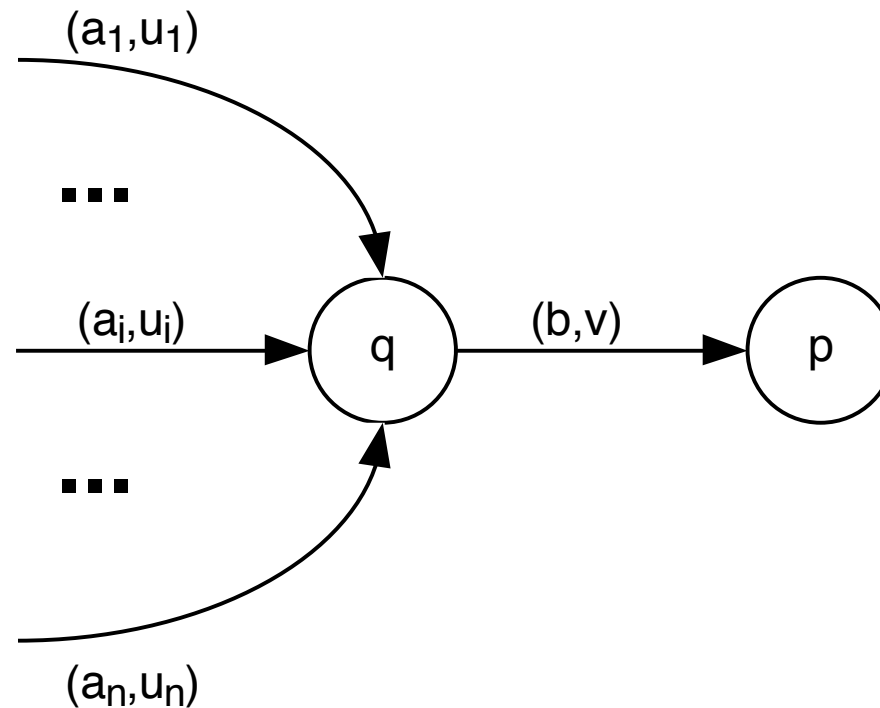
Indução em  $n > 0$  prova que, de fato, ME (Mealy) simula M (Moore)

- ao reconhecer a entrada  $a_1 \dots a_n$
- se M passa pelos estados  $q_0, q_1, \dots, q_n$
- e gera as saídas  $u_0, u_1, \dots, u_n$ ,
- então ME passa pelos estados  $q_e, q_0, q_1, \dots, q_n$
- e gera as saídas  $u_0 u_1, \dots, u_n$

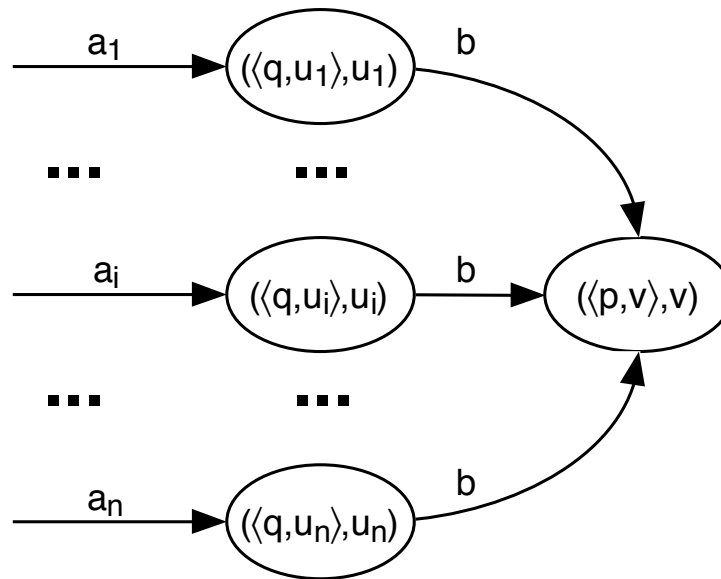
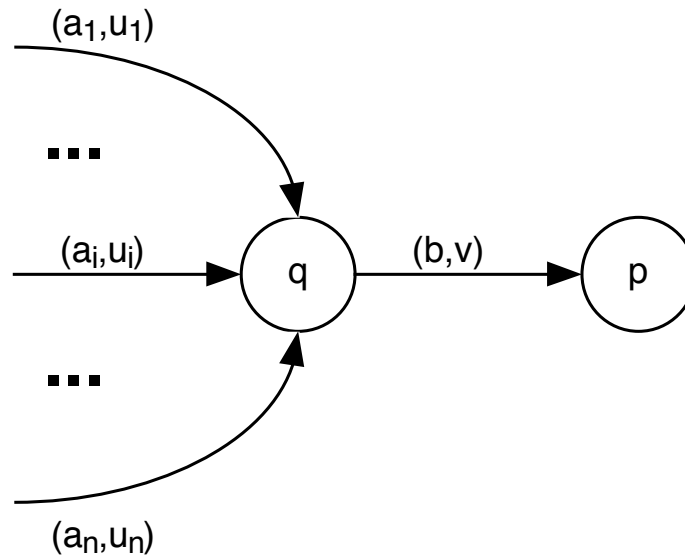
## Teorema: Máquina de Mealy $\rightarrow$ Máquina de Moore

Toda Máquina de Mealy pode ser simulada por uma Máquina de Moore

Prova: (por indução)



Correspondente Moore ???



## Correspondente Máquina de Moore

- em geral, **mais estados** que Mealy
- transições com **saídas diferentes** atingem um **mesmo estado**
  - \* simulado por **diversos estados** (um para cada saída)
  - \* **estado** é um par ordenado  $\langle \text{estado}, \text{saída} \rangle$

$M = (\Sigma, Q, \delta, q_0, F, \Delta)$ , Mealy qualquer. Correspondente Moore

$$MO = (\Sigma, (Q \times S(\delta)) \cup \{ \langle q_0, \epsilon \rangle \}, \delta_{MO}, \langle q_0, \epsilon \rangle, F \times S(\delta), \Delta, \delta_S)$$

- $S(\delta)$ : imagem de  $\delta$ , restrita à componente saída
  - \* conjunto de saídas possíveis de  $M$
- se  $\delta(q_0, a) = (q, u)$

$$\delta_{MO}(\langle q_0, \epsilon \rangle, a) = \langle q, u \rangle$$

- se  $\delta(q, b) = (p, v)$ , então, para cada  $\delta(q_i, a_i) = (q, u_i)$

$$\delta_{MO}(\langle q, u_i \rangle, b) = \langle p, v \rangle$$

- para o estado  $\langle q, u \rangle$  de  $MO$

$$\delta_S(\langle q, u \rangle) = u$$

Indução em  $n$  prova que, ao reconhecer a entrada  $a_1 \dots a_n$

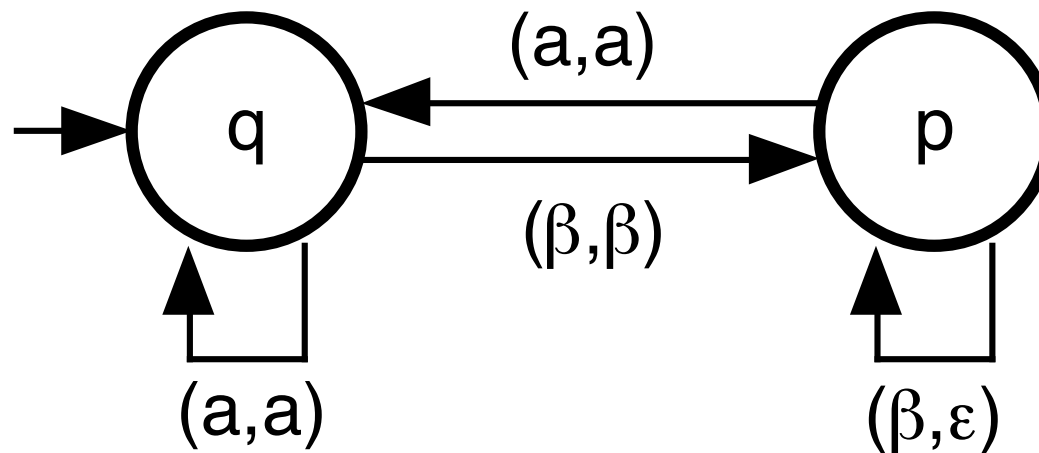
- se  $M$  passa pelos estados  $q_0, q_1, \dots, q_n$
- e gera as saídas  $u_1, \dots, u_n$
- então  $MO$  passa pelos estados  $\langle q_0, \epsilon \rangle, \langle q_1, u_1 \rangle, \dots, \langle q_n, u_n \rangle$
- e gera as saídas  $\epsilon, u_1, \dots, u_n$

## Exp: Máquina de Mealy $\rightarrow$ Máquina de Moore

$$M = (\{ a, \beta \}, \{ q, p \}, \delta, q, \{ q, p \}, \{ a, \beta \})$$

Máquina de Mealy

- compacta brancos de um texto

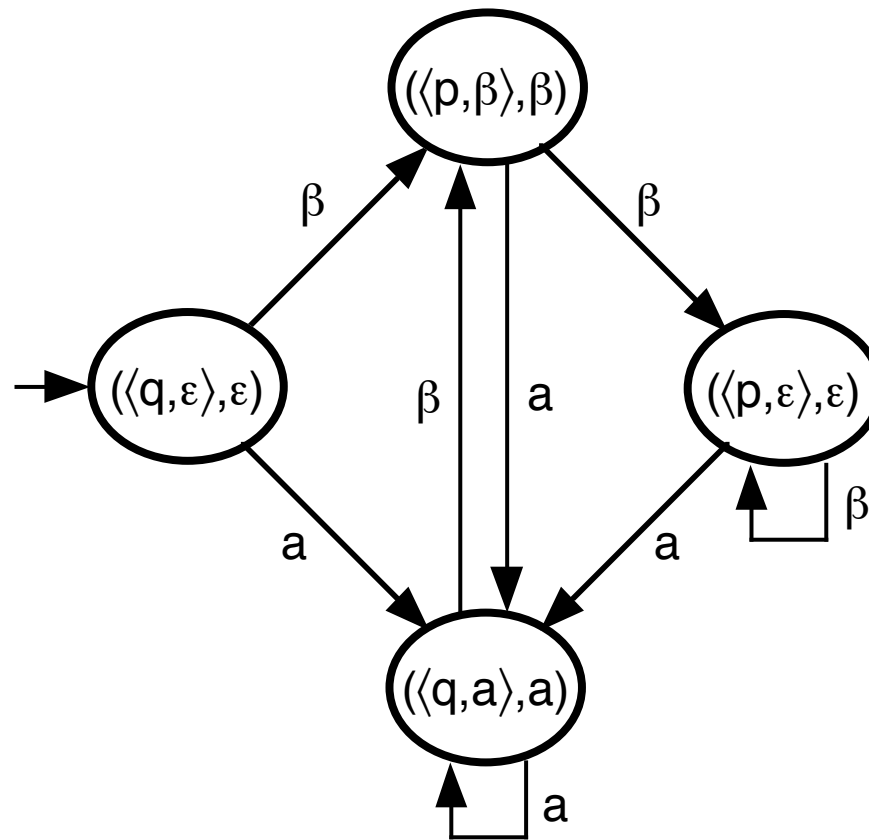




$MO = (\{ a, \beta \}, Q, \delta_{MO}, \langle q, \epsilon \rangle, F, \{ a, \beta \}, \delta_S)$

Máquina de Moore

- $Q = F = \{ q, p \} \times \{ \epsilon, a, \beta \}$



## Obs: Máquina de Mealy × Máquina de Moore

Mealy possui, em geral

- menos estados que a correspondente Moore

Em aplicações práticas, sempre que possível,

- usar Mealy preferencialmente a Moore

Em experimentos reais, significativa preferência das pessoas

- associar as saídas aos estados (e não às transições).
- sugere-se especial atenção a este fato

# 5 – Autômato Finito com Saída

5.1 Máquina de Mealy

5.2 Máquina de Moore

5.3 Equivalência das Máquina de Moore e Mealy

5.4 Hipertexto e Hiperímia como Autômato Finito com Saída

5.5 Animação como Autômato Finito com Saída

## 5.4 Hipertexto e Hiperímia como Autômato Finito com Saída

### ◆ Hipertexto

- **ponteiros** ou **links** entre diversas **páginas**
- **texto** possui “**âncoras**” que **apontam** para **páginas** do documento

### ◆ Hiperímia

- **extensão** desta noção para recursos **multímia**
  - \* imagens, animações e sons

## ◆ Noção de hipertexto

- proposta por Vannevar Bush em 1945, objetivando
  - \* armazenar uma grande quantidade de documentos
  - \* interligados de acordo com uma semântica de associação
  - \* flexibilizando/otimizando tempos de recuperação de informações

## ◆ Associação hipertexto/hipermídia à WWW

- documentos com ponteiros fisicamente codificados nas páginas
- tal solução compromete
  - \* reusabilidade e atualização dos recursos usados

## ◆ Idealmente, hipertexto (hipermídia) deve possuir

- estrutura navegacional independente dos dados sobre a qual é construído

## ◆ Hipertextos (hipermídias) vistos como AF com saída

- alfabeto de entrada: conjunto de rótulos dos ponteiros
  - \* modificações no alfabeto?
- função programa: estrutura navegacional
  - \* determina a estruturação lógica
  - \* modificações na função programa?
- alfabeto de saída: conjunto de recursos hipertexto/hipermídia
  - \* armazenados na base de dados
  - \* modificações no alfabeto de saída?
- palavra de saída: uma página, composta por símbolos do alfabeto de saída (recursos hipertexto/hipermídia) “concatenados”
  - \* modificações nas saídas?

## ◆ Resultado

- páginas e ponteiros de um hipertexto/hipermídia em um sítio
- cada autômato com saída: visão da mesma base de dados

## ◆ Hipermídia vista como um autômato finito com saída

- pode possuir restrições nos tempos/sincronizações entre mídias
  - \* decorrentes das limitações de expressividade das LR
  - \* limitações sobre o que os autômatos finitos podem computar

## Exp: Hiperdocumento como Autômato Finito com Saída

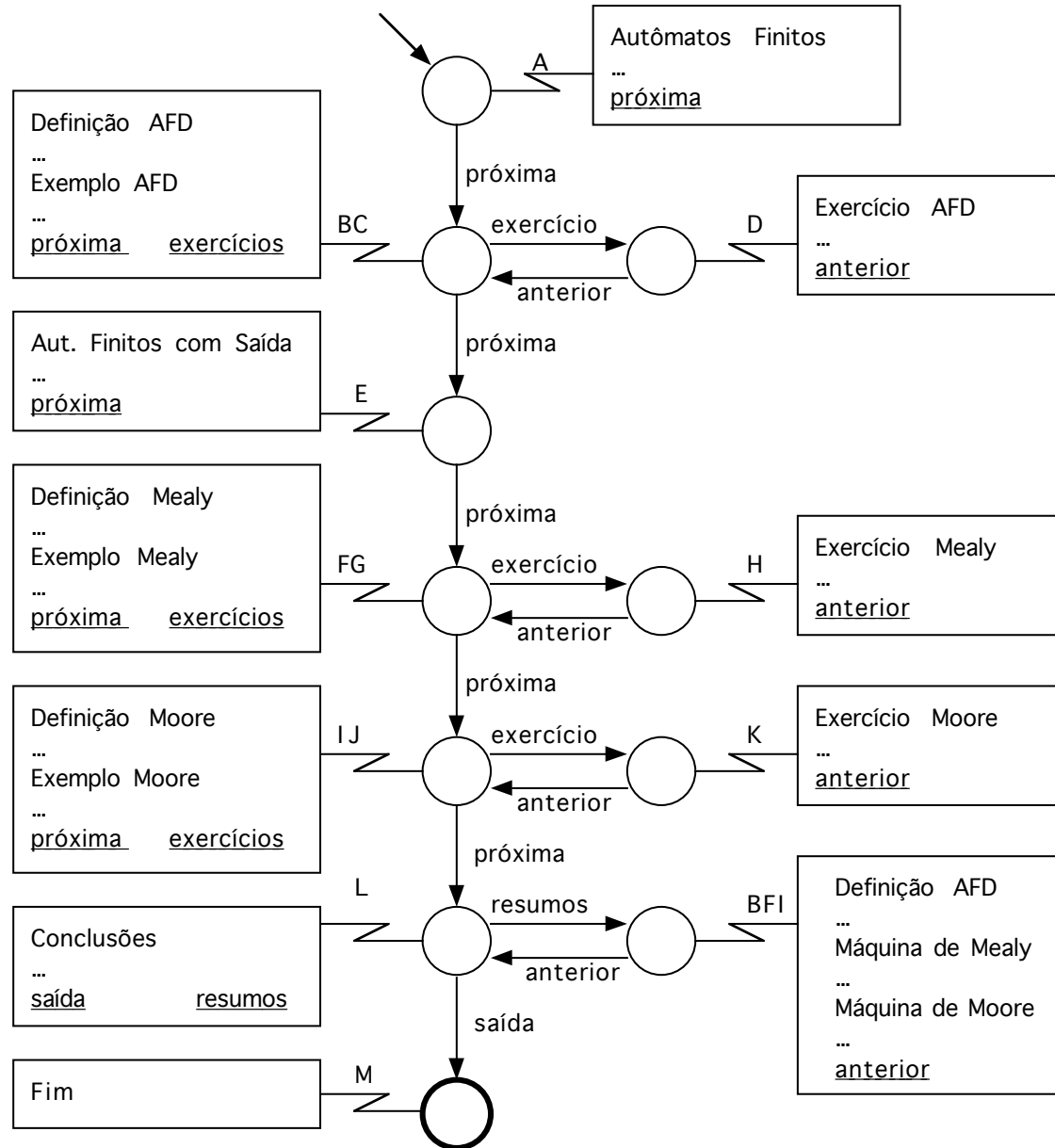
Hipertexto com objetivo de disponibilizar um [Curso](#) sobre [Autômatos com Saída](#) na [WWW](#), usando Máquina de [Moore](#)

Alfabeto de entrada: { próxima, exercício, anterior, resumos, saída }

Alfabeto de saída: { A, B, C, D, E, F, G, H, I, J, K, L, M }

- **A** - Introdução AF
- **B** - Definição AFD
- **C** - Exemplo AFD
- **D** - Exercício AFD
- **E** - Introdução AF com Saída
- **F** - Definição Máquina de Mealy
- **G** - Exemplo Máquina de Mealy
- **H** - Exercício Máquina de Mealy
- **I** - Definição Máquina de Moore
- **J** - Exemplo Máquina de Moore
- **K** - Exercício Máquina de Moore
- **L** - Conclusões
- **M** - Fim





## ◆ Observe

- fragmentos de hipertextos são concatenados, compondo páginas
- mesmos fragmentos são usados em mais de uma página
  - \* reuso de fragmentos de hipertextos
- se um fragmento for alterado na base de dados
  - \* todas referências são automaticamente alteradas no autômato
- símbolos do alfabeto de entrada são rótulos de ponteiros

## ◆ Exercício

- Máquina de Mealy

## ◆ Vantagens

- base de dados
  - \* alto grau de modularização dos recursos
  - \* facilidade de reuso desses recursos
- independência da estrutura navegacional (programa) do conteúdo
  - \* modificações na estrutura navegacional
  - \* não influem no conteúdo (e vice-versa)
- facilidade
  - \* criação/manutenção de hipertextos/hipermídias
  - \* criação de hipertexto/hipermídia sobre algum já existente
- interface gráfica simples e direta (AF como diagrama)
- implementação trivial

## ◆ Exercício: não-determinismo

- interpretação no contexto de hipertextos/hipermídias na WWW??

# 5 – Autômato Finito com Saída

**5.1 Máquina de Mealy**

**5.2 Máquina de Moore**

**5.3 Equivalência das Máquina de Moore e Mealy**

**5.4 Hipertexto e Hiperímia como Autômato Finito com Saída**

**5.5 Animação como Autômato Finito com Saída**

## 5.5 Animação como AF com Saída

### ◆ Sistemas de animação para

- criação
- apresentação de animações

### ◆ Podem ser

- Tempo real
  - \* imagem exibida é computada no momento da visualização
- Quadro-a-quadro
  - \* imagem exibida é previamente computada e armazenada

## ◆ *World Wide Web*

- sistemas de **animação** são **especialmente importantes**
- grande parte de seu conteúdo contém animações

## ◆ **Questões importantes**

- taxa de **transmissão**
- **espaço** de armazenamento
- **tempo** de processamento

## ◆ Sistemas de animação quadro-a-quadro na WWW

- AVI - Audio Video Interleave
- MPEG - Moving Picture Expert Group
- QuickTime
- GIF - Graphics Interchange Format

## ◆ Características desejáveis de um sistema de animação

- reutilização
  - \* seqüências de imagens
  - \* partes específicas de imagens
  - \* para compor animações a partir de animações existentes
- busca de informações (principalmente em animações complexas)
  - \* ocorrência de determinadas condições ao longo da animação



## ◆ Animações quadro-a-quadro vistas como AF c/ saída

- cada autômato: um ator
- composição de atores em camadas: animações

## ◆ Cada ator

- fita de entrada independente
- alfabeto de saída: conjunto de imagens e sons elementares do ator
- palavra de saída
  - \* imagem / som do ator
  - \* a cada instante da animação
- alfabeto de entrada: conjunto de ações possíveis
- função programa: comportamento do ator

## ◆ Desejável estender o modelo com facilidades específicas para animações

- controle de tempos
- transformações aplicadas a imagem ou som

## ◆ Uma solução: célula de fita de entrada é uma tripla

- símbolo do alfabeto de entrada
- tempo de processamento da transição (exibição da imagem)
- transformação aplicada
  - \* imagem: posicionamento, rotação, etc.
  - \* som: volume, equalização, etc.

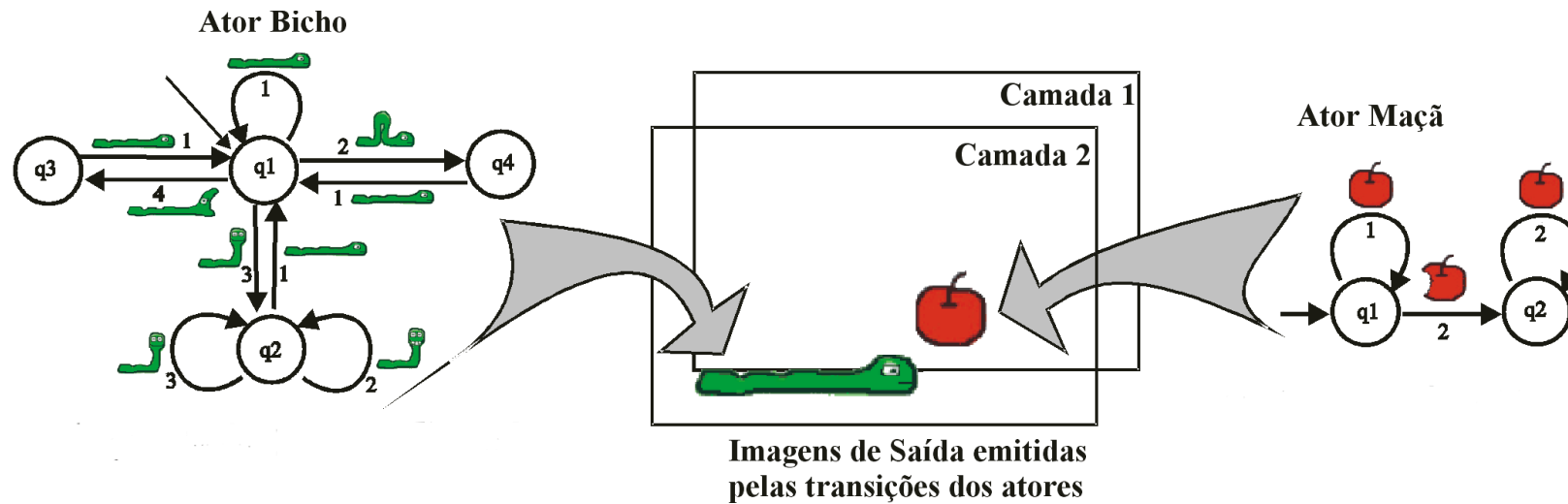
# Exp: Animação como AF com Saída

## Atores

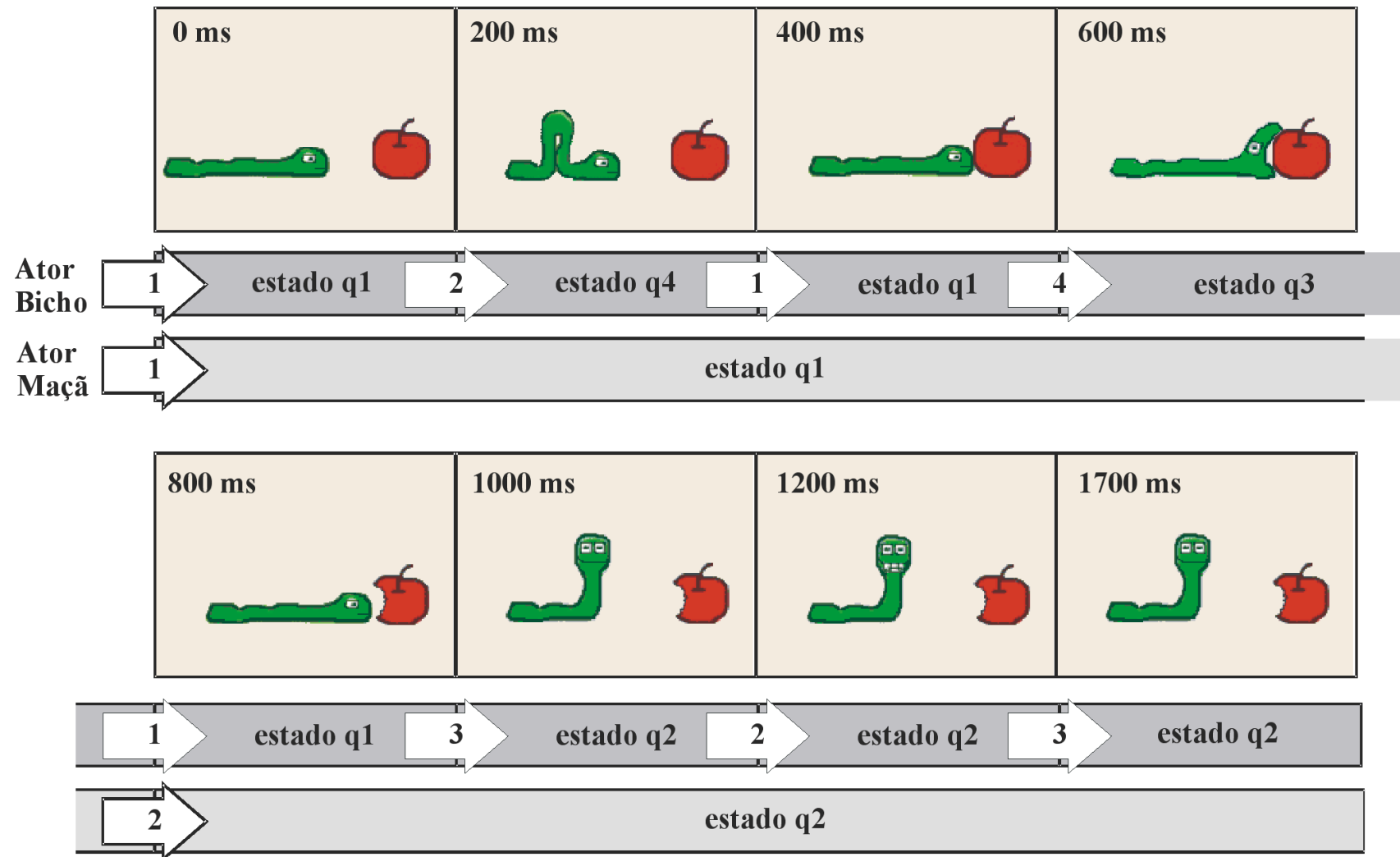
- **cobra** capaz de se movimentar, abocanhar e rir
- **maçã** que pode estar ou não mordida

## Animação

- **cobra** eventualmente **abocanha** a **maçã** (Máquina de **Mealy**)



- imagens dos atores: **camadas**, compondo **quadros** das **animações**



- **sincronização** dos atores: controle de **tempos**

## ◆ Observe que

- alteração algum símbolo do alfabeto de saída (imagem elementar)
  - \* todas referências são automaticamente alteradas
- ator pode ser reusado na composição de uma outra animação
- mesmo ator pode ser usado diversas vezes em uma animação
  - \* exemplo: animação com diversas cobras independentes

## Exercício

- Máquina de Moore

## ◆ Vantagens

- encapsulamento das propriedades estéticas e comportamentais em uma unidade básica (ator) favorece
  - \* reuso (instanciação) em diferentes animações
  - \* existe apenas *um* autômato (e diversas fitas de entrada)
- independência da estrutura comportamental (programa) do conteúdo das imagens/sons
  - \* modificações na estrutura comportamental
  - \* *não* influem no conteúdo das imagens/sons (e vice-versa)
- facilidade
  - \* criação e manutenção de atores e animações
  - \* criação de ator/animação sobre algum já existente
- interface gráfica simples e direta (AF como diagrama)
- implementação trivial

## ◆ Importante vantagem (animações complexas)

*buscas de informações sobre a ocorrência de determinadas condições ao longo de uma animação*

- usando
  - \* estrutura de estados
  - \* algumas informações adicionais



## ◆ Comparação com modelos usuais quadro-a-quadro

- importante vantagem: tamanho de arquivo (taxa de transferência)
- pode montar cada quadro no momento em que é exibido
  - \* mesma imagem exibida em diferentes momentos da animação
  - \* *sem* necessidade de codificar (ou transmitir) novamente
- o mesmo para
  - \* diferentes instâncias do mesmo ator
  - \* diferentes atores que usam o mesmo alfabeto de saída

## ◆ Casos reais

- 20% ou menos do espaço usualmente requerido por um GIF

## ◆ Exercício: não-determinismo

- interpretação no contexto de animações???

# Linguagens Formais e Autômatos

P. Blauth Menezes

- 1 **Introdução e Conceitos Básicos**
- 2 **Linguagens e Gramáticas**
- 3 **Linguagens Regulares**
- 4 **Propriedades das Linguagens Regulares**
- 5 **Autômato Finito com Saída**
- 6 **Linguagens Livres do Contexto**
- 7 **Propriedades e Reconhecimento das Linguagens Livres do Contexto**
- 8 **Linguagens Recursivamente Enumeráveis e Sensíveis ao Contexto**
- 9 **Hierarquia de Classes e Linguagens e Conclusões**

# Linguagens Formais e Autômatos

**P. Blauth Menezes**

blauth@inf.ufrgs.br

**Departamento de Informática Teórica  
Instituto de Informática / UFRGS**

